11-03-00

A

FISH & RICHARDSON P.C.

4350 La Jolla Village Drive
Suite 500
San Diego, California
92122

October 31, 2000

Telephone
858 678-5070

Facsimile
858 678-5099

Web Site
www.fr.com
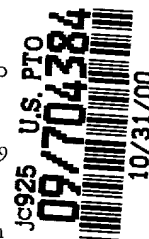
Attorney Docket No.: 10559-299001

Box Patent Application
Commissioner for Patents
Washington, DC 20231

Presented for filing is a new original patent application of:

Applicant: VICTOR B. LORTZ, YLIAN SAINT-HILAIRE AND JAMES L. JASON JR.

Title: NETWORK POLICY DISTRIBUTION

Enclosed are the following papers, including those required to receive a filing date under 37 CFR §1.53(b):

	Pages
Specification	9
Claims	6
Abstract	1
Declaration	2
Drawing(s)	8

Enclosures:
— Postcard.

Basic filing fee	$710
Total claims in excess of 20 times $18	$180
Independent claims in excess of 3 times $80	$400
Fee for multiple dependent claims	$0
Total filing fee:	$1290

Frederick P. Fish
1855-1930

W.K. Richardson
1859-1951

BOSTON
DALLAS
DELAWARE
NEW YORK
SAN DIEGO
SILICON VALLEY
TWIN CITIES
WASHINGTON, DC

FISH & RICHARDSON P.C.

A check for the filing fee is enclosed. Please apply any other required fees or any credits to deposit account 06-1050, referencing the attorney docket number shown above.
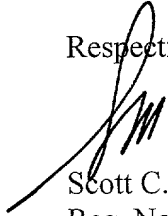
If this application is found to be incomplete, or if a telephone conference would otherwise be helpful, please call the undersigned at (617) 542-5070.

Kindly acknowledge receipt of this application by returning the enclosed postcard.

Please send all correspondence to:

SCOTT C. HARRIS
Fish & Richardson P.C.
4350 La Jolla Village
Suite 500
San Diego, California 92122

Respectfully submitted,

Scott C. Harris
Reg. No. 32,030
Enclosures
SCH/laf
20148777 doc

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: NETWORK POLICY DISTRIBUTION

APPLICANT: VICTOR B. LORTZ, YLIAN SAINT-HILAIRE AND JAMES L. JASON JR.

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No _____ EL558604216US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C 20231

_____ 10/31/00 _____

Date of Deposit

_____ Derek Norwood _____

Signature

_____ Derek W. Norwood _____

Typed or Printed Name of Person Signing Certificate

NETWORK POLICY DISTRIBUTION

TECHNICAL FIELD

This invention relates to network policy distribution.

BACKGROUND

For two computers to communicate with each other over a
network, the network systems on the two computers typically
are configured so that they are compatible. For example, if
two computers are to encrypt data communicated between them,
both computers may use the same encryption and decryption
methods. For small networks with simple computers, an
operator may manually configure the network systems at each of
the computers.

Alternatively, a policy server computer on the network
may configure the network systems of the computers on the
network. The network configuration data for the computers are
stored in policy files or in a policy database on the policy
server. The policy server configures the network systems of
the computers by transmitting configuration data from the
policy files or database to the computers. Policy clients on
the computers may then change the network configuration of the
computers using the transmitted configuration data.

For a policy client to configure a network system using a
policy file, the policy file is typically formatted in a
particular way. The rules for formatting a policy file are
referred to as a schema. Different clients may require
different policy files formatted according to different
schemas.

DESCRIPTION OF DRAWINGS

FIG. 1 is a block diagram of a network of computers
according to the invention;

FIG. 2 shows a first part of a network policy according
to a first version;

FIG. 2 shows a first part of a network policy according to a first version;

FIG. 3 shows a schema of the first version of the network policy;

FIG. 4 shows a schema of a second version of a network policy;

FIG. 5A shows a second part of the network policy of FIG. 2 and a translation specification;

FIG. 5B shows a translated version of the network policy of FIG. 1;

FIG. 6 is a flow chart of the process implemented by the server of FIG. 1;

FIG. 7A is a flow chart of the process implemented by one of the clients of FIG. 1; and

FIG. 7B is a flow chart of the process implemented by another one of the clients of FIG. 1.

DETAILED DESCRIPTION

As shown in FIG. 1, a network 10 connects client computers 12, 14, 16 to a server 18. The server 18 has a processor 20 for executing computer programs, a network interface 24 for communicating over the network 10, and a storage subsystem 22. Storage subsystem 22 may include a CDROM drive, floppy disk, hard disk, hard disk array, memory, and so on. Processor 20 may cause the network clients 12-16 to change their network configurations by executing policy server 26 software, stored within storage subsystem 22.

Policy server software 26 is associated with a network policy 30, which includes a particular version 34 of a network configuration 32 and a timestamp 35 indicating the time that the network configuration 32 was created. The network configuration 32 is used to configure the clients 12, 14, 16. The clients 12, 14, 16 may differ from each other, for example, because they are from different manufacturers or have different operating systems. Consequently the clients may require different versions of the network configuration 32.

2

The policy 30 includes a specification 36 for translating the network configuration 32 from the particular version 34 to another version of the policy that may be required by the clients 12, 14, 16. The policy 30 may contain multiple translation specifications 36 to allow the policy to be translated into many different versions.

The policy server software 26 also includes a policy transmitter 40 that transmits the policy 30 over the network 10 to the client computers 12, 14, 16 and a policy translator 42 that can be used to translate the network configuration 32 using the translation specification 36 for client computers 16 that cannot do the translation.

The client computers 12, 14, 16 each contain a network interface 52a, 52b, 52c for communicating with the network 10, a storage subsystem 53a, 53b, 53c, and a processor 54a, 54b, 54c for executing software stored within the storage subsystem. The software sends and receives data over the network through a network system 56a, 56b, 56c and the network interface 52a, 52b, 52c. Policy client software 58a, 58b, 58c, stored within the storage subsystem 53a, 53b, 53c configures the network system 56a, 56b, 56c using the network policy 30 transmitted by the server 18.

Policy client software 58a, 58b, 58c includes a network configurator 60a, 60b, 60c that receives the policy 30 from the server 18 and uses the policy 30 to configure the network system 56a, 56b, 56c. Network configurator 60a, 60b, 60c also records a timestamp 57a, 57b, 57c associated with the network policy 30 that was used to configure the network system 56a, 56b, 56c in the network system. Each network configurator 60a, 60b, 60c is associated with a particular ``schema'' 62a, 62b, 62c of the network configuration 32. A network configurator may not be able to configure a network interface with a policy file that is configured according to a different schema from the one that is expected by the network configurator. The schemas 62a, 62b, 62c of the network

clients are assigned version numbers 64a, 64b, 64c to distinguish one schema from another.

Clients 12, 14 have a policy translator 70a, 70b, which uses the translation specification 36 contained within the policy 30 to translate the network configuration 32 contained within the policy 30 to the version 64a of the schema 62a, 62b associated with the client 12, 14. The network configurator 60a, 60b then configures the network system 56a, 56b using the translated policy. Thus, sending the translation specification 36 to the clients 12, 14 along with the network configuration allows the same policy 30 to be used in configuring clients that may have different policy versions. Since only one policy 30 is maintained on the server, the amount of effort and resources needed to maintain the policy 30 is reduced. Traffic within the network 10 may also be reduced by broadcasting the policy 30 to both clients 12, 14 in the same transmission.

Client 16 is not equipped with a policy translator. If the policy 30 does not have the same version 64c as client 16, client 16 sends a policy error 71 to the server 18. The policy translator 42 on the server 18 then translates the policy 30 into the version 64c associated with the client 16 and transmits the translated policy 72 to the client 16. The network configurator 60c of the client 16 uses the translated policy 72 to configure the network system 56c. Thus the policy 30 can be used with a client 16 that is not equipped to translate the policy.

The policy 30 may, for example, be represented in a tag-based language such as the eXtensible Markup Language (XML), HyperText Markup Language (HTML), or Standard Generalized Markup Language (SGML). The invention will be described with reference to an implementation where the policy is represented as an XML file with the translation specification being represented in an eXtensible Stylesheet Language (XSL) file. The policy translators 40, 70a, 70b are eXtensible Stylesheet language translators (XSLT). XML and XSL are document

4

formatting and translating languages promulgated by the World Wide Web Consortium (W3C). XSL translators are easily available because XSL is used in many different applications. For example, the XSL translator included in Internet Explorer by Microsoft Inc. may be used as a policy translator 40, 70a, 70b.

As shown in FIG. 2, a network policy 30 specifies the version 34 of a schema to which the policy 30 conforms and a timestamp 35 indicating the creation time of the policy. Policy 30 conforms to a schema version 1.1. Policy 30 has network configuration data 32 for configuring network systems 56a, 56b, 56c (FIG. 1) of client computers. Network configuration data 32 includes network data encryption settings 90 that network systems 56a, 56b, 56c (FIG. 1) may use when communicating with another computer. The network systems 56a, 56b, 56c of a client computer 12, 14, 16 configured according to the policy 30 proposes a first network setting 90a as an encryption setting for communication between the client computer 12, 14, 16 and another computer. If the first setting 90a is unacceptable to the other computer, the network system 56a, 56b, 56c propose a second setting 90b to the other computer. The first setting 90a proposes a pre-shared 92a authentication method with a DES 94a cipher algorithm, an MD5 hash algorithm 96a and a group identity 98a of DH768. The second setting 90b proposes a ``DSS signatures'' 92b authentication method with a ``Rjindael'' 94b cipher algorithm, an SHA-1 hash algorithm 96b and a group identity 98b of DH1024.

As shown in FIG. 3, the schema 100 for the policy 30 (FIG. 2) lists the requirements 101 for the attributes of a network-setting proposal in the policy 30. According to the schema, an ID 102 is required 104 for each proposal. A cipher algorithm 108 selected from ``3DES,'' ``DES,'' ``IDEA,'' ``Blowfish,'' ``RC5,'' ``CAST,'' and ``Rjindael'' 110 is also required. A hash algorithm 114, a Group ID 116, and an authentication method 118 respectively selected from lists

120, 122, and 124 are also required in the network setting proposal. Schema 100 specifies an arrangement of a version ``1.1'' policy file that can be used by the network configurator 60a, 60b, 60c. A network configurator may not be

5 able to configure a network system with a policy file that does not conform to the schema of the network configurator.

As shown in FIG. 4, an older schema 130 associated with version ``1.0'' network policies does not include the ``Rjindael'' algorithm 94b of policy 30 (FIG. 2) in the list

10 132 of cipher algorithms 134. Consequently, a network configurator associated with the older schema 130 cannot configure a network system using policy 30.

As shown in FIG. 5A, an XSL translation specification 36 (see also FIG. 1) may be used to translate policy 30 to an

15 older version ``1.0'' which conforms to the older schema 130. The specification includes a set 142 of statements that replace instances of the ``Rjindael'' cipher algorithm with the ``3DES'' algorithm, which is defined in the older schema 130. A statement 142a in the set 142 directs a policy translator to

20 check whether a policy has ``Rjindael'' defined as a cipher algorithm and another statement 142b directs policy translator to change the cipher algorithm to change the cipher algorithm from ``Rjindael'' to ``3DES.'' Thus the specification 36 directs a policy translator to translate a policy from version

25 1.1 to version 1.0. A statement 146 directs the policy translator to change the policy so that it reflects the older schema version 1.0.

FIG. 5B shows the translated policy 72 resulting from translating policy 30 (FIG. 2) based on specification 36. As

30 shown in FIG. 5B, the version number 150 has been changed from ``1.1'' to ``1.0'' and the cipher algorithm 160 has been changed from ``Rjindael'' to ``3DES.'' However, the timestamp 35 of the policy 72 has not been changed because the translation process does not affect the creation date of the

35 policy. The translated policy 72 may be used by clients 12,

14, 16 (FIG 1) that require a policy that conforms to the schema 130 of FIG. 4.

5

10

15

20

As shown in FIG. 6, the server 18 begins the process of configuring the clients 12, 14, 16 by broadcasting (600) the network policy 30 on the network 10, so that the policy 30 is received by all the clients 12, 14, 16. By sending the policy 30 to the clients 12, 14, 16 in a single transmission, the server reduces the network traffic needed to configure the clients. The server 18 then checks (602) whether a policy error 71 has been received from any of the client computers 12, 14, 16. If a policy error 71 has not been received, the server 18 continues to wait for an error 71.

Otherwise the server 18 determines (604) the address of the client computer 16 that sent the policy error 71 from the information contained within the policy error. The server 18 also extracts (606) the version 64c of the schema 62c of the client from the information contained within the policy error 71. Upon extracting the version 64c, the policy translator 40 on the server 18 translates (608) the network configuration 32 of the policy 30 to the version 64c using the translation specification 36 to produce a translated policy 72. The server transmits (610) the translated policy 72 to the client computer 16 and stops the process.

25

30

35

As shown in FIG. 7A, the process of configuring a client 12, 14 that includes a policy translator 70a, 70b begins when the client 12, 14 receives (700) a policy 30 from the server 18. The client 12, 14 extracts (701) the timestamp 35 from the policy 30 and checks (702) whether the timestamp 35 is greater than the timestamp 57a, 57b of a network policy that was previously used to configure the network systems 56a, 56b of the clients. A greater timestamp 35 indicates that the policy 30 was created more recently than the policy that was previously used to configure the network systems 56a, 56b of the clients. If the timestamp 35 is not greater, the client 12, 14, terminates the configuration process.

Otherwise, if the timestamp 35 is greater, the client 12, 14 extracts (703) a schema version 34 from the policy 30 and checks (704) if the extracted schema version 34 is the same as the version 64a, 64b associated with the client 12, 14. If the extracted schema version 34 is the same as the version 64a, 64b of the client 12, 14, the network configurator 60a, 60b of the client 12, 14 configures (710) the network system 56a, 56b of the client using the policy 30.

Otherwise, if the extracted schema version 34 is not the same as the version 64a, 64b of the client 12, 14, the policy client 58a, 58b extracts (706) the translation specification 36 from the policy 30 and translates (708) the network configuration 32 of the policy 30 using the extracted specification 36. The network configurator 60a, 60b of the client 12, 14 then configures (710) the network system 56a, 56b of the client using the translated policy.

As shown in FIG. 7B, the process of configuring a client 16 that does not have a policy translator also begins when the client 16 receives (749) a policy 30 from the server 18. The client extracts (750) a timestamp 35 from the policy 30, and checks (751) whether the timestamp 35 is greater than the timestamp 57c of a network policy that was previously used to configure the network systems 56c of the client 16. If the timestamp 35 is not greater, the client 16 terminates the process.

Otherwise, if the timestamp 35 is greater, the client 16 extracts (752) a schema version 34 from the policy 30 and checks (754) if the extracted schema version 34 is the same as the version 64c associated with the client 16. If the extracted schema version 34 is the same as the version 64c of the client 16, the network configurator 60c of the client 16 configures (760) the network system 56c of the client using the policy 30 and terminates the process.

Otherwise, if the extracted schema version 34 is not the same as the version 64c of the client 16, the client 16 waits (756) for a predetermined waiting period and then checks (758)

8

whether a translated policy 72 has been received from the server 18. By waiting for the predetermined period, the client 16, can receive a translated policy that was broadcast or transmitted (610 FIG. 6) from the server to another client

5   (not shown) which also does not have a policy translator. The different clients are configured to wait for different predetermined periods so that client computers with longer waiting periods can configure their network systems with translated policies which are sent by the server in response

10   to policy errors from clients with shorter waiting periods. This reduces the number of policy errors and translated policies sent over the network, thereby reducing the network traffic.

If a translated policy 72 has not been received, the

15   policy client 58c sends (759) a policy error 71 to the server 18 and then checks (758) if a translated policy has been received. Otherwise, if a translated policy 72 has been received, the client 16 extracts 750 a timestamp from the translated policy and repeats the process (751-760) described

20   above.

Other embodiments are within the scope of the following claims. For example, the policy 30 and its translation specification 36 may not be sent in a single file. Instead, the policy and the specification may be sent in two separate

25   files, allowing the same specification 36 to be used with two different policies 30. The server 18 may be configured to transmit the translated policy 72 to a single client 16 instead of broadcasting the translated policy 72 to multiple clients in the network 10. In such a single re-transmission

30   network, the configuration process (FIG. 7B) for a client 16 that does not have a policy translator would not include waiting (756) for a delay time before sending (759) a policy error.

WHAT IS CLAIMED IS:

1   1. A method, comprising:
2   receiving a specification for translating a network
3   policy from a first schema to a second, different schema;
4   translating the network policy into the second different
5   schema based on the specification; and
6   configuring a network system based on the translated
7   policy.

1   2. The method of claim 1 wherein the network policy is
2   represented in a tag-based language.

1   3. The method of claim 1 wherein the specification is received
2   in a file from a policy server.

1   4. The method of claim 3 wherein the file also contains the
2   policy.

1   5. An article comprising a machine-readable medium which stores
2   machine-executable instructions for checking events
3   performed by a device, the instructions causing a machine
4   to:
5   receive a specification for translating a policy from a
6   first schema to a second different schema;
7   translate the network policy into the second different
8   schema based on the specification; and
9   configure a network system based on the translated
10   policy.

1   6. The article of claim 5 wherein the network policy is
2   represented in eXtensible Markup Language and the
3   specification is represented in eXtensible Stylesheet
4   Language.

1   7. The article of claim 5 wherein the specification is received
2   in a file from a policy server.

1   8. The article of claim 5 wherein the file also contains the

2   policy.

1   9. An apparatus comprising:

2   a memory which stores computer readable instructions; and

3   a processor which executes the computer readable

4   instructions to:

5   receive a specification for translating a policy

6   from a first schema to a second, different schema;

7   translate the network policy into the second

8   different schema based on the specification; and

9   configure a network system based on the translated

10   policy.

1   10. The apparatus of claim 9 wherein the network policy is

2   represented in eXtensible Markup Language and the

3   specification is represented in eXtensible Stylesheet

4   Language.

1   11. The apparatus of claim 9 wherein the specification is

2   received in a file from a policy server.

1   12. The apparatus of claim 9 wherein the file also contains

2   the policy.

1

1   13. A method, comprising:

2   storing a network policy for configuring a network system

3   according to a first schema;

4   storing a specification for translating the network

5   policy from the first schema to a second different schema;

6   translating the network policy into the second different

7   schema based on the specification; and

8   sending the translated network policy to a client

9   computer.

1   14.   The method of claim 13, further comprising:

2   prior to translating the network policy:

3   sending the network policy to the client computer;

4   sending the specification for translating the

5   network policy to the client computer; and

6   receiving an indication that the client computer

7   cannot translate the network policy.

1   15.   The method of claim 13 wherein the network policy is

2   represented in eXtensible Markup Language and the

3   specification is represented in eXtensible Stylesheet

4   Language.

1   16.   The method of claim 13 wherein the network policy and the

2   specification are stored in one file.

1   17.   An article comprising a computer-readable medium which

2   stores computer-executable instructions for checking events

3   performed by a device, the instructions causing a machine

4   to:

5   store a network policy for configuring a network system

6   according to a first schema;

7   store a specification for translating the network policy

8   from the first schema to a second different schema;

9   translate the network policy into the second different

10   schema based on the specification; and

11   send the translated network policy to a client computer.

1   18.   The article of claim 17, wherein the instructions further

2   cause the machine to:

3   prior to translating the network policy:

4   send the network policy to the client computer;

5   send the specification for translating the network

6   policy to the client computer; and

7   receive an indication that the client computer

8   cannot translate the network policy.

1   19. The article of claim 17 wherein the network policy is
2   represented in eXtensible Markup Language and the
3   specification is represented in eXtensible Stylesheet
4   Language.

1   20. The article of claim 17 wherein the network policy and
2   the specification are stored in one file.

1   21. An apparatus comprising:
2   a memory which stores computer readable instructions;
3   a processor which executes the computer readable
4   instructions to:
5   store a network policy for configuring a network
6   system according to a first schema;
7   store a specification for translating the network policy
8   from the first schema to a second different schema;
9   translate the network policy into the second different
10   schema based on the specification; and
11   send the translated network policy to a client computer.

1   22. The apparatus of claim 21 wherein, prior to translating
2   the network policy, the processor executes the instructions
3   to:
4   send the network policy to the client computer;
5   send the specification for translating the network
6   policy to the client computer; and
7   receive an indication that the client computer
8   cannot translate the network policy.

1   23. The apparatus of claim 21 wherein the network policy is
2   represented in eXtensible Markup Language and the
3   specification is represented in eXtensible Stylesheet
4   Language.

1   24. The apparatus of claim 21 wherein the network policy and
2   the specification are stored in one file.

1   25.   A method of configuring a network comprising:

2   transmitting a network policy according to a first schema

3   and a specification for translating the network policy from

4   the first schema to a second different schema from a server;

5   receiving the network policy and the specification on a

6   first client computer;

7   translating on the client computer the network policy

8   from the first schema to the second different schema using

9   the specification; and

10   configuring the network system on the first client

11   computer using on the translated network policy.

1   26.   The method of claim 25 further comprising:

2   receiving the network policy on a second client computer;

3   and

4   configuring the network system on the second client

5   computer using on the network policy.

1   27.   The method of claim 25 further comprising:

2   receiving the network policy on a third client computer;

3   transmitting to the server an indication that the third

4   client computer cannot translate the network policy;

5   translating on the server the network policy from the

6   first schema to the second different schema using the

7   specification; and

8   transmitting the translated network policy to the third

9   client computer.

1   28.   The method of claim 27 wherein the network policy is

2   represented in eXtensible Markup Language and the

3   specification is represented in eXtensible Stylesheet

4   Language.

1   29.   The method of claim 27 wherein the network policy and the

2   specification are contained in one file.

1   30.   A method of creating a file for configuring a network

2   system comprising:

3   adding network data to the file; and
4   adding a specification for translating the network data
5   from a first schema to a second schema.

ABSTRACT

A method includes receiving a specification for translating a network policy from a first schema to a second, different schema and translating the network policy into the second different schema based on the specification. A network
5   system is configured based on the translated policy.

20128271.doc

CLIENT 12

PROCESSOR — 54a

STORAGE SUBSYSTEM 53a

POLICY CLIENT 58a

POLICY TRANSLATOR 70a

NETWORK 60a CONFIGURATOR

SCHEMA 62a

VERSION 64a

NETWORK SYSTEM — 56a
TIMESTAMP — 57a

NETWORK INTERFACE — 52a

CLIENT 14

PROCESSOR — 54b

STORAGE SUBSYSTEM 53b

POLICY CLIENT 58b

POLICY TRANSLATOR 70b

NETWORK 60b CONFIGURATOR

SCHEMA 62b

VERSION 64b

NETWORK SYSTEM — 56b
TIMESTAMP — 57b

NETWORK INTERFACE — 52b

CLIENT 12

CLIENT 14

POLICY 30

POLICY 30

NETWORK 10

POLICY 30

POLICY ERROR 71

TRANSLATED POLICY 72

POLICY ERROR 71

TRANSLATED POLICY 72

SERVER 18

CLIENT 16

SERVER 18 — 20

PROCESSOR

NETWORK INTERFACE — 24

STORAGE SUBSYSTEM 22

POLICY SERVER 26

POLICY 30

VERSION 34

TIMESTAMP 35

NETWORK 32 CONFIGURATION

TRANSLATION 36 SPECIFICATION

POLICY TRANSMITTER 40

POLICY TRANSLATOR 42

CLIENT 16 — 54c

PROCESSOR

NETWORK INTERFACE — 52c

NETWORK SYSTEM — 56c
TIMESTAMP — 57c

STORAGE SUBSYSTEM 53c

POLICY CLIENT 58c

NETWORK 60c CONFIGURATOR

SCHEMA 62c

VERSION 64c

Fig. 1

// Policy that conforms to the newer schema (specifies Rjindael):

`<?xml version="1.0"?>`

`<!DOCTYPE IPsecPolicyOnSystem PUBLIC "//somestandard.org/IPsecPolicy/Version1.1">` — 34

`<IPsecPolicyOnSystem timestamp="2000-9-30_12:12:12">` — 35

   `<IKEProposals>`

     90a `<IKEProposal id="Proposal1" CipherAlgorithm="DES" HashAlgorithm="MD5" GroupId="DH768" AuthenticationMethod="Preshared"/>`

     90b `<IKEProposal id="Proposal2" CipherAlgorithm="Rjindael" HashAlgorithm="SHA-1" GroupId="DH1024" AuthenticationMethod="DSS_Signatures"/>`

   `</IKEProposals>`

`</IPsecPolicyOnSystem>`

90

92a
92b
94a
94b
96a
96b
98a
98b

30

Fig. 2

```
// DTD schema Version 1.1 for IPsec implementations that support the new
// encryption algorithm called Rjindael:

<?xml version="1.0" encoding="UTF-16"?>

<!ELEMENT IKEProposal EMPTY>

<!ATTLIST IKEProposal
    id ID #REQUIRED                                                    — 104    — 162
                                                    110
    CipherAlgorithm (3DES | DES | IDEA | Blowfish | RC5 | CAST | Rjindael) #REQUIRED    — 168    — 112
                                    120
    HashAlgorithm (MD5 | SHA-1 | Tiger) #REQUIRED                      — 114
                                    122
    GroupId (DH768 | DH1024 | DH1536 | ECC2N155 | ECC2N185) #REQUIRED  — 116
    AuthenticationMethod (DSS_Signatures | Preshared | RSA_Signatures |    — 118
        RSA_Encryption | Revised_RSA_Encryption | Kerberos) #REQUIRED    124
>
```

101

100

```
<!ELEMENT IKEProposals (IKEProposal+)>

<!ELEMENT IPsecPolicyOnSystem (IKEProposals)>

<!ATTLIST IpsecPolicyOnSystem
    timestamp CDATA #REQUIRED                                          — 126
>
```

Fig. 3

```
// DTD schema for version 1.0 IPsec policy implementations (doesn't
// include the new encryption algorithm called Rjindael):

<?xml version="1.0" encoding="UTF-16"?>

<!ELEMENT IKEProposal EMPTY>

<!ATTLIST IKEProposal

    id ID #REQUIRED

    CipherAlgorithm (3DES | DES | IDEA | Blowfish | RC5 | CAST) #REQUIRED

    HashAlgorithm (MD5 | SHA-1 | Tiger) #REQUIRED

    GroupId (DH768 | DH1024 | DH1536 | ECC2N155 | ECC2N185) #REQUIRED

    AuthenticationMethod (DSS_Signatures | Preshared | RSA_Signatures |

    RSA_Encryption | Revised_RSA_Encryption | Kerberos) #REQUIRED

>

<!ELEMENT IKEProposals (IKEProposal+)>

<!ELEMENT IPsecPolicyOnSystem (IKEProposals)>

<!ATTLIST IPsecPolicyOnSystem

    timestamp CDATA #REQUIRED

>
```

132

13y

130

Fig. 4

```
// XSLT file to transform the above policy into a policy conforming to the older
// schema. This transformation simply replaces attribute "Rjindael" with "3DES".

<?xml version="1.0" encoding="UTF-8"?>
<!-- This stylesheet replaces the CipherAlgorithm Rjindael with 3DES, for IKE
implementations that do not support Rjindael. -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>

    <xsl:output doctype-public="//somestandard.org/IPsecPolicy/Version1.0"/>  ⟋ 146

        <xsl:template match="/*">
        <ᴀsl:element name="IPsecPolicyOnSystem">
        <xsl:copy-of select="@*"/>
        <xsl:element name="IKEProposals">
        <xsl:apply-templates select="IKEProposals"/>
        </xsl:element>
        </xsl:element>
        </xsl:template>

    <xsl:template match="IKEProposals/IKEProposal">

        <xsl:element name="IKEProposal">
        ⎧  <xsl:copy-of select="@*"/>
        ⎪
        ⎪   <xsl:if test="@CipherAlgorithm = 'Rjindael'">  ⟋ 142a
        ⎨
        ⎪       <xsl:attribute name="CipherAlgorithm">3DES</xsl:attribute>  ⟋ 142b
        ⎪
        ⎩   </xsl:if>
    142

        </xsl:element>

    </xsl:template>

</xsl:stylesheet>
```

⟋ 36

Fig. 5A

// Output resulting from applying the above transformation to the policy:

```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE IPsecPolicyOnSystem PUBLIC "//some.standard.org/IPsecPolicy/Version1.0">

<IPsecPolicyOnSystem timestamp="2000-9-30_12:12:12">    ~35

<IKEProposals>

  <IKEProposal id="Proposal1" CipherAlgorithm="DES" HashAlgorithm="MD5" GroupId="DH768" AuthenticationMethod="Preshared"/>
                                                                       ~160
  <IKEProposal id="Proposal2" CipherAlgorithm="3DES" HashAlgorithm="SHA-1" GroupId="DH1024" AuthenticationMethod="DSS_Signatures"/>

</IKEProposals>

</IPsecPolicyOnSystem>
```
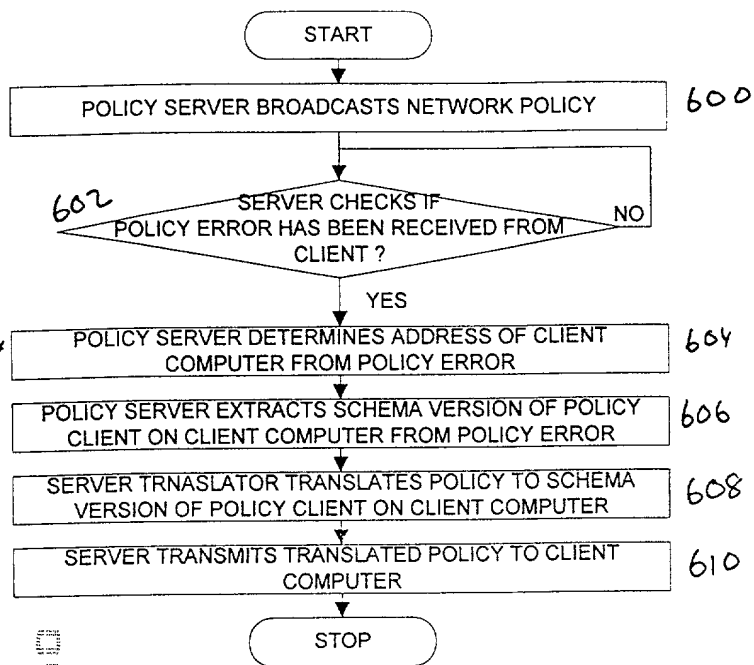
~150

~12

Fig. 5B

START

| POLICY SERVER BROADCASTS NETWORK POLICY | 600 |

602 SERVER CHECKS IF
POLICY ERROR HAS BEEN RECEIVED FROM   NO
CLIENT ?

YES

| POLICY SERVER DETERMINES ADDRESS OF CLIENT COMPUTER FROM POLICY ERROR | 604 |

| POLICY SERVER EXTRACTS SCHEMA VERSION OF POLICY CLIENT ON CLIENT COMPUTER FROM POLICY ERROR | 606 |

| SERVER TRNASLATOR TRANSLATES POLICY TO SCHEMA VERSION OF POLICY CLIENT ON CLIENT COMPUTER | 608 |

| SERVER TRANSMITS TRANSLATED POLICY TO CLIENT COMPUTER | 610 |

STOP

FIG. 6

START

| RECEIVE POLICY .   FROM SERVER | 700 |

| EXTRACT TIMESTAMP FROM POLICY | 701 |

NO   CHECK IF   702
TIMESTAMP > TIMESTAMP OF CURRENT
POLICY OF CLIENT?

YES

| EXTRACT VERSION OF POLICY | 703 |

704 CHECK IF VERSION
OF POLICY IS THE SAME AS VERSION OF   YES
SCHEMA OF CLIENT?

NO   706

| POLICY CLIENT EXTRACTS SPECIFICATION   FOR TRANSLATING POLICY . | |

| CLIENT TRANSLATOR TRANSLATES POLICY BASED ON THE EXTRACTED SPECIFICATION | 708 |

| CONFIGURE CLIENT NETWORK SYSTEM BASED ON POLICY | 710 |

STOP

FIG. 7A

START

RECEIVE POLICY FROM SERVER — 749

EXTRACT TIMESTAMP FROM POLICY — 750

751 — CHECK IF
TIMESTAMP > TIMESTAMP OF CURRENT
POLICY OF CLIENT?   NO

YES

EXTRACT VERSION OF POLICY — 752

CHECK IF VERSION 754
OF POLICY IS THE SAME AS VERSION OF
SCHEMA OF CLIENT?   YES

760

CONFIGURE CLIENT NETWORK SYSTEM
BASED ON POLICY

NO

WAIT FOR A PREDETERMINED DELAY TIME — 756

STOP

YES

CHECK IF   758
TRANSLATED POLICY RECEIVED FROM
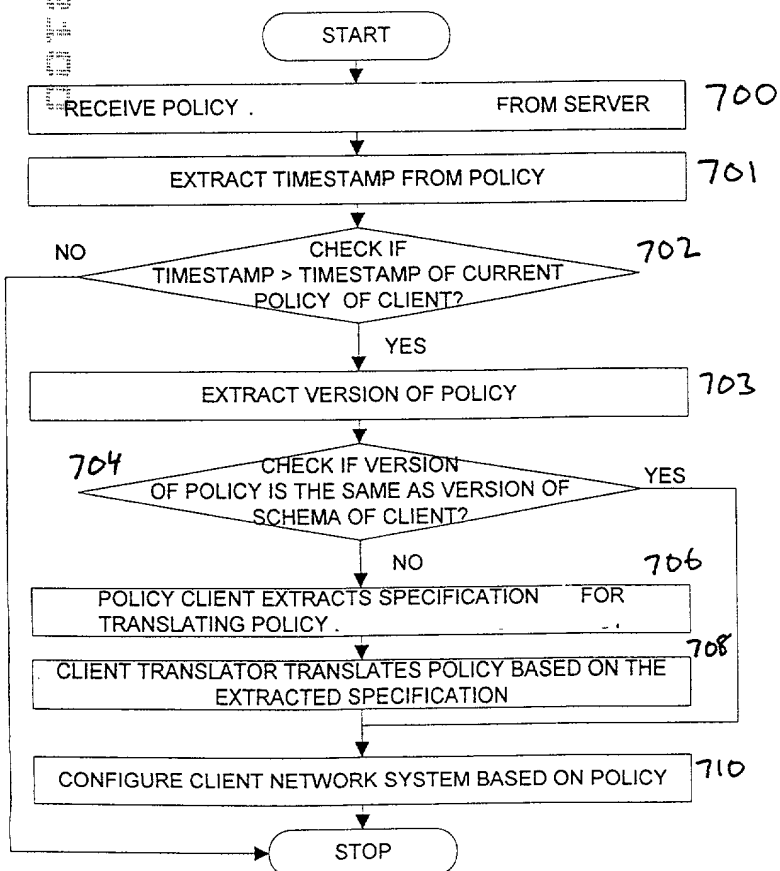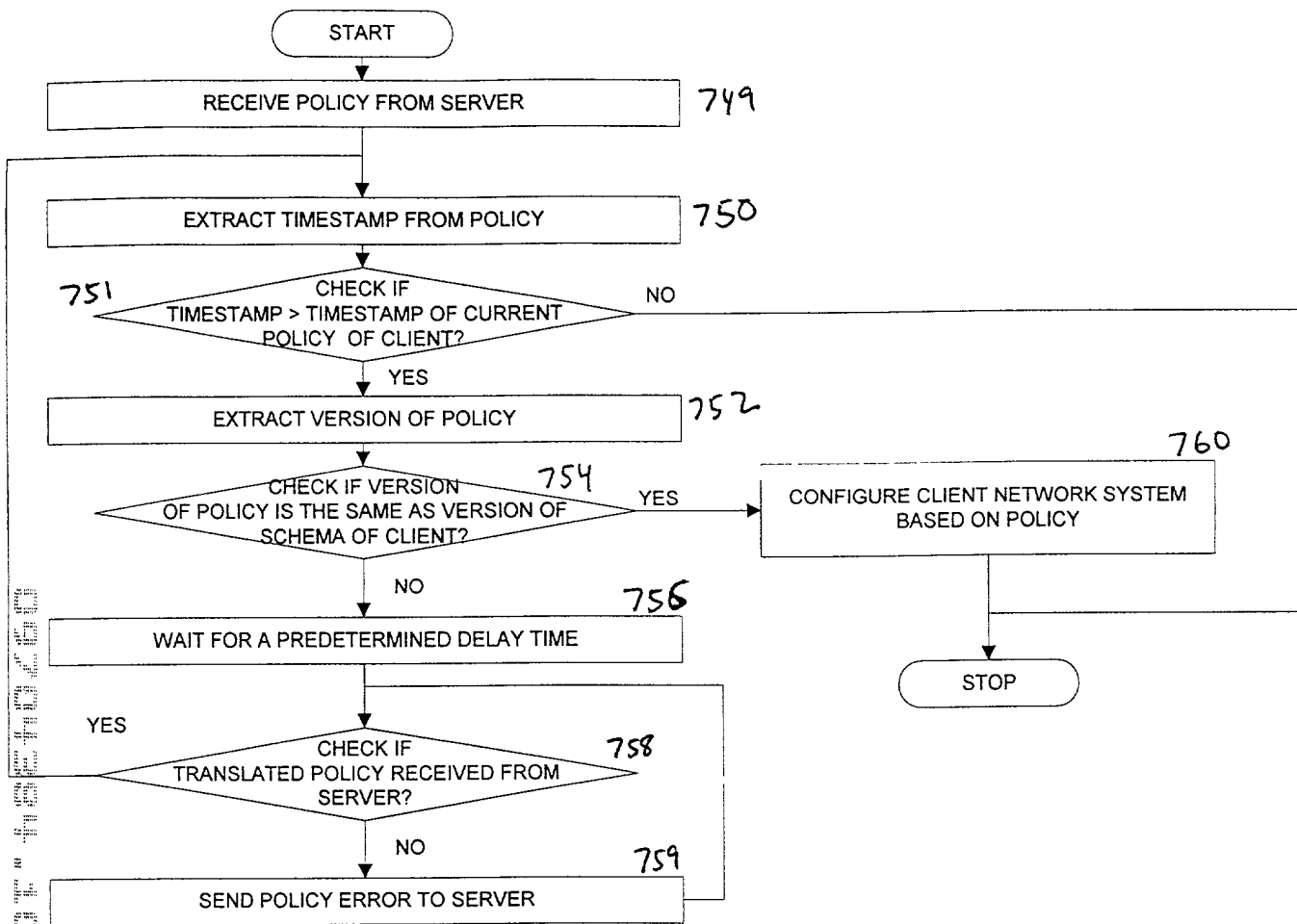SERVER?

NO   759

SEND POLICY ERROR TO SERVER

FIG. 7B

Attorney's Docket No.: 10559-299001
Client's Ref. No.: P9310

COMBINED DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled **NETWORK POLICY DISTRIBUTION**, the specification of which:

[X]   is attached hereto.
[]   was filed on _ as Application Serial No. _ and was amended on _____.
[]   was described and claimed in PCT International Application No. _____ filed on
      _____ and as amended under PCT Article 19 on _____.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose all information I know to be material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby appoint the following attorneys and/or agents to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

David L. Feigenbaum, Reg. No. 30,378   Scott C. Harris, Reg. No. 32,030
John F. Hayden, Reg. No. 37,640   Timothy A. French, Reg. No. 30,175

Address all telephone calls to DAVID L. FEIGENBAUM at telephone number (617) 542-5070.

Address all correspondence to SCOTT C. HARRIS at:

FISH & RICHARDSON P.C.
4350 La Jolla Village
Suite 500
San Diego, California 92122

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Full Name of Inventor: VICTOR B. LORTZ

Inventor's Signature: _____   Date: 10/30/00

Residence Address: ~~2111 N.E. 25th Avenue~~ 7716 S.W. Oviatt Dr.
                         ~~Hillsboro, OR 97124~~ Beaverton, OR 97007

Citizenship: ~~2111 N.E. 25th Avenue~~ USA
                         ~~Hillsboro, OR 97124~~

Post Office Address: ~~2111 N.E. 25th Avenue~~ Same as residence
                         ~~Hillsboro, OR 97124~~

Combined Declaration and Power of Attorney
Page 2 of 2 Pages

Full Name of Inventor: YLIAN SAINT-HILAIRE

Inventor's Signature: _Ylian Saint-Hilaire_   Date: **30 / 10 / 2000**
Residence Address: 1316 NE Carlaby Way #173
                       Hillsboro, OR 97124
Citizenship: Canada
Post Office Address: 1316 NE Carlaby Way #173
                       Hillsboro, OR 97124

Full Name of Inventor: JAMES L. JASON JR.

Inventor's Signature: Date: **30-oct-2000**
Residence Address: 1643 NE Glen Ellen Drive
                       Hillsboro, OR 97124
Citizenship: United States
Post Office Address: 1643 NE Glen Ellen Drive
                       Hillsboro, OR 97124

20157810.doc